



Gould, N ORCID logoORCID: <https://orcid.org/0000-0001-7199-2201> and Mackaness, W (2016) From taxonomies to ontologies: formalizing generalization knowledge for on-demand mapping. *Cartography and Geographic Information Science*, 43 (3). pp. 208-222. ISSN 1050-9844

Downloaded from: <https://e-space.mmu.ac.uk/582556/>

Version: Accepted Version

Publisher: Taylor & Francis

DOI: <https://doi.org/10.1080/15230406.2015.1072737>

Please cite the published version

<https://e-space.mmu.ac.uk>

From taxonomies to ontologies: formalizing generalization knowledge for on-demand mapping

Nicholas Gould^{a*}, William Mackaness^b

^aSchool of Science and the Environment, Manchester Metropolitan University, UK

^bThe School of GeoSciences, University of Edinburgh, UK

Abstract

Automation of the cartographic design process is central to the delivery of bespoke maps via the web. In this paper ontological modelling is used to explicitly represent and articulate the knowledge used in this decision making process. A use case focuses on the visualization of road traffic accident data as a way of illustrating how ontologies provide a framework by which salient and contextual information can be integrated in a meaningful manner. Such systems are in anticipation of web based services in which the user knows what they need, but do not have the cartographic ability to get what they want.

Keywords: ontologies, map generalization, cartographic design, OWL, web services

Introduction

The democratization of cartography refers to the idea that the creator of the map and the users of the map are one and the same (Morrison 1997). But in the absence of any cartographic training, the technology does indeed allow the masses to make ‘cartographic monstrosities with unprecedented ease’ (Monmonier 1984, p389). The response from research communities has been to develop sophisticated algorithms and methodologies by which the cartographic design process is embedded as a service within web based environments – thus obviating the need for cartographically aware users. This web based service is referred to as ‘on-demand’ mapping – a service in which users can create, in real time, tailor made maps, by combining data from multiple sources. The challenge in providing such a service is in formalizing the knowledge necessary to support the complex process of design. What is required is a modelling of the underpinning geography sufficient to be able to support the decision making process of map design. Ontological modelling (Dutton and Edwardes, 2006) holds great promise in making explicit various cartographic conceptualizations, and thus providing a means of governing that cartographic process – a process that very much reflects a compromise among various (sometimes competing) objectives.

On Demand Mapping

On-demand mapping is defined as, ‘the creation of a cartographic product upon a user request appropriate to its scale and purpose’ (Cecconi 2003, p17). The emphasis here being on ‘On-demand’ mapping rather than ‘on the fly’ where performance issues would be paramount. The vision is of a web-based map production environment that facilitates production of multi-scaled thematic maps that suit a particular user’s needs; the idea being that a map specific to a

* Corresponding author. Email: n.gould@mmu.ac.uk

given task is likely to be more readily understood than using a generic multi-purpose map (Wilson, Bertolotto, and Weakliam 2010). The ambition of on-demand mapping has been hugely facilitated by the various open access initiatives of National Mapping Agencies (e.g. Ordnance Survey's OpenData initiative). Cartographic design is known to be a highly complex decision making process. If we assume that an on-demand mapping system will be utilized by non-cartographers then the process of selecting, sequencing and execution of generalization algorithms needs to be concealed from the user.

Our contention is that high levels of automation in on-demand mapping can only be realized if we are able to incorporate generalization knowledge, and take account of geographic features and their relationships. Such concepts and relationships can be formalized (and shared) through the use of ontologies (Yan et al, 2015; Balley and Regnauld, 2011; Dutton and Edwards 2006).

Currently the cartographic knowledge required in map generalization lies implicit among various cartometric and map generalization algorithms, often embedded in complex mapping systems (Taillandier and Taillandier 2012), or in the mind of the cartographer. Such systems are unwieldy; their knowledge base requiring update each time user specifications change (Taillandier and Taillandier 2012). The advent of on-demand mapping, and the desire to map *any* feature type (not just the features of the multi-purpose topographic map), have exacerbated the problem. This can be overcome through the use of ontologies which have a number of distinct benefits (Burger and Simperl, 2008) not least of which is the ability for domain experts to share their knowledge. Ontological modelling is seen as a way of making explicit such knowledge in order to develop on-demand systems capable of generating output at a wide range of scales and across a rich palette of themes. We would also argue that by representing knowledge in a *declarative* form (as opposed to *procedural*) we specifically allow *machines* to reason about geographic space and its representation. This in turn can provide a meaningful way by which users can reason with the machine. The remainder of the paper is organized as follows; we begin with a review of methodologies for representing cartographic generalization knowledge. The next section provides a justification for representing this knowledge in an ontology and introduces a methodology for building an ontology. How the ontology was built and implemented in an on-demand mapping prototype is described in the next section. The paper concludes in the last section with a summary of the main findings and suggestions of further work.

Representing cartographic knowledge

To automate generalization and move away from 'batch' systems, rule-based expert systems were developed to encapsulate generalization knowledge in a machine-readable form. For example, Nickerson (1991) attempted to convert the guidelines for producing the Canadian National Topographic Series into condition-action rules. A number of problems with the rule-based approach have been identified, not least the sheer number of rules required (Harrie and Weibel 2007); the cascading effects of generalization, which rule-based systems find difficult to control; and the problem of competition between rules (Fisher and Mackaness 1987). Beard (1991) highlighted the challenges of dealing with multiple user needs from a

finite set of rules. Many of these problems stem from the condition-action rules used to represent procedural knowledge. Procedural knowledge describes ‘what to do, when’ (Rich and Knight, 1991, p113); there is a coupling of the knowledge with a description of what to do with that knowledge. This has its advantages but procedural knowledge is hard to repurpose. A rule is required for every situation and in a dynamic system, such as on-demand mapping, the rule-based approach is unfeasible.

The *constraints* concept, first proposed by Beard (1991) solves some of the problems of rule-based system by providing a separation of conflict *analysis* and conflict *solution* (Burghardt, Schmid, and Stoter 2007): the constraint defines the desired output and an optimization technique determines the best combination of generalization operators to meet that constraint. An example of a constraint might be: at 1:50K the minimum size of a building is 50m². At their simplest, constraints represent *declarative* knowledge. Declarative knowledge makes statements, or declarations, about the world (Genesereth and Nilsson, 1998). Constraints might, of course conflict: a minimum size constraint on two neighboring buildings, which could lead to an increase in size, might conflict with an overlap constraint (on all features), which could lead to a displacement, which might then conflict with a positional constraint. However, this is not the concern of the constraints; it is the role of the optimization technique to resolve such conflicts. This is a feature of declarative knowledge; it needs a system to decide what to do with the knowledge (Rich and Knight 1991). A number of optimization techniques have been developed but the agent-based method (Lamy et al. 1999) is the only one that can utilize a range of generalization operators (Harrie and Weibel 2007) and has seen extensive use in map production environments (Revell, Regnauld, and Bulbrooke 2011; Lecordix and Lemarie 2007). Given its success in automated map production, why not use the constraints model for on-demand mapping? Its success has been in the production of maps to particular scales, 1:25K, 1:50K etc., but constraints such as those that represent minimum size or minimum separation require threshold values that might not suite all map conditions.

It is also true that sophisticated workflow models have been used to derive smaller scale maps from large scale detailed data. Again, these developments have been scale specific (e.g. 1:25K, 1:50K), and had topographic maps as their focus (Stoter et al. 2014). But our focus is on the production of thematic maps, at a range of varying levels of detail. Furthermore, we contend that to realize on-demand mapping, it is necessary to represent the semantics of generalization and that constraints do not achieve this. If we can describe *why* a constraint is necessary then that knowledge can be repurposed.

If we see on-demand mapping as a knowledge engineering problem then how is cartographic knowledge represented in generalization systems? It can exist implicitly and informally within the cartographer’s mind (lower left quadrant - Figure 1). Most research has focused on formalizing this knowledge (making it machine readable), though it remains implicit – making it difficult to transfer or reuse (upper left quadrant). Some attempts have been made to make explicit cartographic knowledge through the use of taxonomies (lower right quadrant). A taxonomy can be regarded as an ontology without semantics (Kavouras and

Kokla 2008). These attempts at taxonomies reveal the subjective nature surrounding the definition and use of generalization operators.

[Figure 1 near here]

Figure 2 shows three generalization operator taxonomies (Roth, Brewer, and Styker 2011; Foerster, Stoter, and Köbben 2007; McMaster and Shea 1992) with equivalent concepts linked by lines. This example reveals how taxonomies frequently use different terms to describe the same operators (synonymy), and in some cases the same term is used to describe different operators (polysemy). There are also differences in the scope of the operators. For example, McMaster and Shea define a *Refinement* operator that has the same role as two operators - *Elimination* and *Typification* - in the Foerster et al. taxonomy.

[Figure 2 near here]

There are various reasons for these ‘taxonomic diversions’ (Kavouras and Kokla (2008). However, accounting for these differences does not necessarily help to resolve them and even if an agreed taxonomy and description of operators were devised it would not be sufficient for an automated system. These taxonomies describe the operators using natural language and although explicit, are still informal (Figure 1). To allow for reasoning and the automatic selection of operators and algorithms we require formal, machine-readable descriptions of operators rather than natural language descriptions (Kilpeläinen 2000). This can be achieved by encapsulating cartographic knowledge in an ontology (Upper right quadrant - Figure 1).

Building an Ontology

An ontology can be defined as a ‘formal, explicit specification of a shared conceptualization’ (Studer, Benjamins, and Fensel 1998, p184). The key term is conceptualization - described by Genesereth and Nilsson (1998) as ‘the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them’. They describe a conceptualization as an abstract, simplified view of the world, which is why conceptualizations are partial, imprecise, and conflicting (Kavouras and Kokla 2008). In our case we are trying to create an abstraction (the ontology) of a process (generalization) that produces an abstraction (a map). This explicit representation of knowledge enables us to reason about the generalization process. Given the complexity of this task it is not surprising that differences, such as those described earlier, arise.

This is not the first time that ontologies have been used in map generalization research. For example Kulik, Duckham, and Egenhofer (2005) used ontologies to aid road line simplification; by Dutton and Edwardes (2006) to represent the roles of geographic features and semantic and structural relationships between features in a coastal region; by Wolf (2009) to influence the aggregation and dimensional collapse of features; and by Lüscher, Weibel, and Mackaness (2008) to aid the recognition of terrace houses. However, none of these ontologies sought to describe the process of generalization; the identification, sequencing and execution of generalization algorithms.

The proposed ontology can be classed as an *application* or *task* ontology, rather than a *domain* ontology whose purpose is to ‘fix’ the knowledge in a particular domain. There are number of ontology design methodologies (Iqbal et al. 2013), however none of these are appropriate for the design of an *application* ontology. Therefore, a hybrid methodology was used (Figure 3) drawing from elements of a number of methodologies (De Nicola, Missikoff, and Navigli 2009; Noy and McGuinness 2001; Uschold et al. 1998).

[Figure 3 near here]

The ontology was implemented in the Web Ontology Language (OWL) using the Protégé ontology editor (Stanford Center for Biomedical Research 2014). Protégé implements a number of logical reasoners that can be used to draw inferences from the assertions made in the ontology. The methodology includes a feedback loop in case the evaluation phase identifies any problems with the design. Whether any redesign occurs at the informal, semi-formal or formal phases depends on the extent of the inadequacies. The more serious the problems the further back in the process it may be necessary to go. We now consider how each phase of the methodology was executed.

Developing an ontology for mapping road accidents

Defining the scope

A use case is necessary, particularly for an application ontology, since it drives the exploration of the application area (De Nicola, Missikoff, and Navigli 2009). A use case serves to limit the scope of the ontology; here our focus is on mapping road accidents. There are various contexts in which we might require such information to be displayed at different scales (Table 1). Any given output involves mapping non-topographic, thematic features (the accidents) along with topographic features (the underlying road network) that share the same geographic space. The aim of mapping user-supplied data with data that might be supplied by a National Mapping Agency can be regarded as a typical application of on-demand mapping. The scope of the ontology is restricted by the application of the use case and the ontology, in its first incarnation, should only contain sufficient knowledge to support that use case; otherwise the ontology becomes inordinately complex.

However, this does not imply a sparse ontology. A road accident has many characteristics: an urban or rural location, the season, the day of the week, the time of day, the state of the road, the road’s classification and capacity, what other things were involved in the collision, and its severity. The road accident can be viewed as an event, physically constrained by the underlying road network. The network too has multiple dimensions: consisting as it does of sections of roads of different classification and shape, which may or may not be one-directional; linked by junctions of different types such as roundabouts, which may or may not be controlled by traffic lights. Any given event can be described within this multi-dimensional space; a description that is scale dependent. If we want to understand the accident from the driver’s perspective we require a fine level of detail. If we are looking for broad patterns across the whole network, we need to look at the data at the synoptic small scale. Across the continuum from the very detailed to the highly synoptic (Table 1), we are

required to make a set of cartographic compromises between arriving at clarity through abstraction of the salient (in this case, road accident data), and including sufficient contextual information so as to give meaning to the data being portrayed (i.e., the underlying road network). In order to manage all of these characteristics we have found it useful to identify a set of map users, each requiring a map at a different scale (Table 1).

[Table 1 near here]

The road safety expert (Table 1a) will be interested in the synoptic view, in order to identify accidents hotspots in the city center. At this scale, there are too many accidents to map individually, and the road network will need to be thinned for legibility by removing the minor roads. However, since many accidents occur at junctions the minor road arms of a junction might provide useful contextual information and should be retained. At a larger scale, the accident map can be used by a parent to identify a safe walking route to school (Table 1b). For example, walking from A to B via C rather than via D might be safer.

From an automated perspective, a model describing the multi-dimensional relationships between salient and contextual data is required in order that we can reason about the cartography. The ambition is a model that enables 1) the automatic selection of contextual data associated with the task, and 2) generalization to be automatically applied in such a way that the patterns and associations appropriate to a particular scale of observation are clearly conveyed.

The sufficiency of the ontology can be gauged through a set of *competency questions*. These are used to evaluate whether the ontology meets requirements and act as a justification for the ontology (Grüninger and Fox 1995). At this stage they are defined at a relatively high level and in natural language form. Examples include: *When should the road network be generalized? How can the road network be generalized? What is the relationship between roads and accidents? What algorithm will implement a particular operator?* The competency questions are refined and formalized as the ontology design progresses; the process is an iterative one.

Informal conceptualization

Once the scope of the ontology has been defined, the concepts of the domain are enumerated. This can be done informally by listing the important concepts and by providing a natural language description of each (Noy and McGuinness 2001). Though the use case helps define some of the concepts, a model of the process of generalization is required, to determine how the knowledge is reasoned with. This conceptual framework (McMaster 1991), or conceptual model (Sarjakoski 2007), is necessary to provide a high-level description of the process. In this instance, the ‘*Why, When and How*’ model of generalization of McMaster and Shea (1992) was found to be a robust approach that engages with the semantics of generalization. Their model seeks to define why generalization is required (for example, *legibility*), when it should be employed (when certain *geometric conditions* such as *congestion* or *imperceptibility* occur) and *how* it should be implemented, using *generalization operators* such as *amalgamation* and *displacement*, which themselves are implemented via

generalization algorithms. For example, at smaller scales (such as Table 1a) the *road accidents*, particularly at road junctions, will be congested. Employing a medical analogy, we can say that conditions such as congestion can be characterized using *symptoms*; for example, a symptom of congestion is a *high feature density*. Just as a thermometer can measure high temperature the model will require *measure algorithms* to measure symptoms such as high feature density. The terms highlighted in *italics* can be added to the informal list of concepts that the ontology should characterize (Table 2).

[Table 2 near here]

Once the list has been created, any concept that appears in the description of another concept can be underlined; this serves as an indicator of a relationship between two concepts. The list of concepts was informed by the use case, the McMaster and Shea model and by the cartographic generalization literature in general. The first iteration of the list was not exhaustive and was modified during other steps in the methodology (Figure 3). The concepts are represented as *objects* when the ontology is formalized. That transformation from concepts to objects starts in the next phase.

Semi-formal conceptualization

The separation, into distinct phases, of the informal and formal description of concepts is a common feature among many ontology design methodologies. However, to allow for a less abrupt transition from the informal to the formal it is useful to add an intermediate semi-formal phase, which consists of ‘mind maps’ or, more exactly, directed graphs (Sure, Staab, and Studer 2009). Some ontology editing tools, such as Protégé, contain components for viewing ontology structures but these can only be used *after* the ontology has been formalized and are not appropriate for this phase of work. Directed graphs can be used initially to depict the subsumption (‘is a’) relationships between concepts or objects. For example, Figure 4 depicts the *feature type* concept and its sub-concepts.

[Figure 4 near here]

The structure of the ontology depends on its intended use. The *CrimeFeatureType* is not required for the use case but is included in Figure 4 to illustrate the fact that we can use the concept of *inheritance* to reduce the amount of knowledge we need to directly represent. We can *assert* the properties of an *EventFeatureType*; for example, we might declare that, unlike topographic features, event features can share the same geographic space. Since the sub-classes of the *EventFeatureType* will inherit its properties, we can *infer* some of the properties of *AccidentFeatureType* and *CrimeFeatureType*. This is a simple example of the ‘new expressions from old’ (Davis, Shrobe, and Szolovits 1993, p18) that inference offers.

The *FeatureType* class hierarchy (Figure 4) consists of a single class and its sub-classes. However, the advantage of representing knowledge in an ontology is that we can define relationships other than subsumption between different classes. For example, we can describe the relationship between the *AccidentFeatureType* and *RoadFeatureType* that was earlier described in the enumeration of concepts. Such relationships, termed *property* relations,

allow us to model any kind of relationship between classes; for instance, the relationships between disparate concepts such as conditions, algorithms, and operators.

Figure 5 is based on the concepts enumerated in Table 2, and seeks to map the top-level concepts and the relationships between them (displayed as labelled edges). A geometric condition, realized in a *ProblemFeatureCollection* (such as the accidents at junction D in Table 1b) is defined by a *Symptom*. The symptom is identified by a *MeasureAlgorithm* and relieved by a *Remedy*. The concept of a remedy was not initially identified in the previous informal conceptualization phase, but was added as an extension based on the medical analogy. Finally, the *Remedy* is performed by an *Operator*, which in turn is implemented by a *TransformationAlgorithm*.

[Figure 5 near here]

Figure 6 depicts a lower level, particular, example of this model, with the top level concepts removed for clarity. So, if the *HighFeatureDensity* symptom is identified in the *FeatureCollection* by a *PointFeatureDensityMeasureAlgorithm* then it can be relieved by, for example, the *FeatureCountReduction* remedy. This remedy is an effect of two Operators, Aggregation and *SelectionByAttribute*, which can be implemented by appropriate algorithms. For example, the clusters of congested accidents have been aggregated into single points in Table 1a.

[Figure 6 near here]

This may seem a rather painstaking process but it is only when the semantics of generalization are captured in the ontology that inference can be used for decision making. The aim is to *assert* as little as possible and to *infer* as much as possible. However, even for seemingly simple use case, it was still necessary to encapsulate a large amount of knowledge in the ontology. The remainder of this section will consider just three of the particular problems faced.

Amalgamation or Merging?

An operator can be implemented by one or more algorithms. The abstract concept of the operator could be omitted and the ontology asked directly to suggest an algorithm that resolves a particular geometric condition. Although the aim of the ontology is to automate the process of generalization and thus the emphasis is on machine understanding, a knowledge representation, such as an ontology, is also a medium for human communication (Kavouras and Kokla 2008). The concept of an operator is well-understood and has been in use for decades and since it is important to reflect the experts' view of the domain in the ontology, the concept was retained.

If the operator concept is to be modelled then how are we to describe its sub-classes? How do we overcome the problem, described earlier (Figure 2), of differing concepts of generalization operators? Should one particular taxonomy be favored over another? This goes against the definition of an ontology as a *shared* conceptualization. Synonymy is not a problem, since if two operator objects from different taxonomies share the same

characteristics but have different names, then that can be managed in the ontology, specifically OWL, by using multiple labels for the object. Indeed, some advocate using a meaningless ID to identify an object whilst maintaining a separate set of terms for that concept (Stevens and Lord 2012). Polysemy can also be managed through the ontological approach since a concept in the ontology is defined by its properties and not its name. So, for example, the two different concepts of Merging, described earlier, can both be modelled but will require different terms. The reasoner will be indifferent to the terms used and, for the human reader of the ontology, metadata can be used to record different concepts. For example that concept A is McMaster and Shea's (1992) view of *Merging* and concept B is the view of Roth et al. (2011). The categories depicted in Figure 2 are for human convenience and can be omitted or included; multiple inheritance allows a class to be a child of multiple categories if necessary. All that matters is for the system to find an algorithm that performs the required transformation.

Class or individual?

A key activity in the semi-formalization phase is determining whether a concept should be modelled as a class or an individual (Noy and McGuinness 2001). *Geometry* has been identified as a key concept: features are defined by their geometry and the application of operators is determined partly by that geometry. For example, it is possible to *simplify* an area or line feature, but not a point feature. Geometry could be defined as a single class with three individuals - *point*, *line* and *area*. Alternatively *point*, *line* and *area* could exist as subclasses of the geometry class and the individuals would be the geometries of particular features or feature collections. Noy and McGuinness (2001) suggest that subclasses should be introduced if they have additional properties to the superclass or if they take part in different relationships from the superclass. They also suggest that if the concept is important, then it should be introduced as a subclass. For this reason the three types of geometry were modelled as subclasses.

A similar problem occurs when we consider how to represent generalization (transformation) algorithms. It might be sensible to represent algorithms as individuals in the operator class since it is difficult to conceive what else the individuals in an operator class might be. However, this would leave no home for the measure algorithms. The algorithm is a sufficiently distinct concept from an operator to justify its representation in a distinct class, with each implementation of an algorithm modelled as an individual in that class.

Modelling spatial relationships

The definition of a road accident as a punctual event that takes place *on* a road network implies a semantic relationship between the two feature types. The nature of the 'on' relationship is vague. Consider the different meanings between 'the nose on your face', 'the house on the street', and 'the box on the floor' (Varanka and Caro 2013). The relation between the accident and the road is similar to the last of these. This semantic relation can be modelled in the ontology as a property (Figure 7).

[Figure 7 near here]

This relationship will allow the on-demand mapping system to infer that when the user wishes to map road accidents then it is appropriate to also map the road network. However, to allow for the generalization of the road network with respect to the accident data, the imprecise semantic relation, ‘is on’, needs to be expressed as a spatial relation. In fact, the semantic relation is expressed as a number of spatial relations which are dependent on the current geometries of the two feature types (Figure 8). For example, if the road accidents have been amalgamated to a polygon then the relationship is *intersects*, but if the road is represented as an aerial feature, and the accidents are not amalgamated, *contained by* is the relationship.

[Figure 8 near here]

Such relations, between objects at the same scale, can be classed as *horizontal relations* (Steiniger and Weibel 2007). It might seem obvious that they should be modelled as properties of classes. (E.g. *accidentFeatureType isContainedBy roadFeatureType*). However, in OWL, properties have limited characteristics. An inverse property can be defined (the inverse of *follows* is *isFollowedBy*) as can a symmetric property (*adjacent*, for example) but there is no way of adding attributes to properties. For example, this would be necessary if we wanted to specify the respective geometries when defining the *isContainedBy* relationship. The solution is to model spatial relations as classes though this is at the expense of making the reasoning process more complex. Figure 9 depicts just one of those relationships, *AccidentContainedByRoad*, which is a sub-class of the more generic *ContainedBy* relationship.

The concept of a *test algorithm* is introduced since it was necessary to test whether a particular accident was contained by a road feature; the implication being that if it is then the road feature should be preserved during generalization.

[Figure 9 near here]

Space does not permit a depiction of all of the graphs developed in the semi-formal conceptualization phase or a full discussion of all of the design decisions made in this phase. Suffice to say that the ontology went through many versions and may yet require further refinement. However, the next step was to formalize these concepts in OWL.

Formal conceptualization

The concepts defined in the directed graphs were encapsulated in OWL-DL, which provides support for Description Logics reasoning, using the Protégé ontology editor. Protégé is a free, open-source, and widely used ontology editor. Classes and subclasses were created in Protégé, based on the directed graphs created in the semi-formal conceptualization phase, after which the object properties (relations) were added. Finally individuals were added. The semi-formal stage can to some extent defer the question of whether a concept exists as a class or as an individual in the ontology. However, the question cannot be avoided in this formal conceptualization phase.

Another advantage of formalizing the ontology using Protégé is that the concept of a *defined* class can be used to check whether two concepts are equivalent. A defined class is where the necessary *and* sufficient conditions for membership have been stated. This is in contrast to a *primitive* class where only the *necessary* conditions have been defined. If the operator sub-classes are marked as defined classes then the reasoner will highlight any that are equivalent. That is, they have the same definition in which case *either* one or both definitions need to be refined *or* one definition is created with two different terms.

The design decisions, made in the *semi-formalization* phase and implemented in the *formalization* phase, are tested in the final, *evaluation* phase.

Evaluating the ontology

The ontology is evaluated, firstly, by asking the competency questions defined in the first phase. For example the competency question: ‘Find an algorithm that will implement a particular operator’ is expressed in the *Manchester OWL syntax* as:

TransformationAlgorithm and *implements* some **Collapse** and *hasInputGeometry* some **AreaGeometry**

where classes are in **bold** and relationships are in *italic*. The query should return a list of TransformationAlgorithm sub-classes and individuals in those classes that meet the specified criteria of implementing the collapse operator on areal features. If any of the competency questions are not answered sufficiently then the ontology needs to be redesigned. For example, congestion in the road accident features can be resolved by the remedy *FeatureCountReduction*. To select an appropriate Operator the following query can be used:

Operator and *hasEffect* some **FeatureCountReduction** and *hasInputGeometry* some **PointGeometry**

The current version of the ontology returns three candidate operators: *selection by attribute* (mapping only the most important features, Figure 10b), *amalgamation* (points to polygon Figure 10c) and *aggregation* (points to point, Figure 10d). All three operators vary in their outputs; selection by attribute retains the concept of individual accidents, and their distinct details, rather than abstract them into clusters, amalgamation retains the geography of the cluster and aggregation provides the best indication of the numeric size of each cluster. So there is scope, in a future version, for refining the operator definitions by including these characteristics.

[Figure 10 near here]

These characteristics could be linked with the requirements of the users listed in Table 1. For example, the *Parent* user would not be interested in individual details of accidents, only in hot-spots, thus the aggregation of accidents would be appropriate.

A comprehensive ontology is not on its own sufficient; a *reasoning agent* is required to make use of the ontology. In particular, since the aim was to develop an *application* ontology to support on-demand mapping then the only satisfactory way of evaluating the ontology is to

develop a prototype on-demand mapping system that will reason with the ontology. Space does not permit a detailed description of the prototype but its design is summarized in Figure 11. Since the system modifies the individuals in the ontology, a copy of it is taken each time the application runs. The *Mapping Engine* component interacts with the ontology via the OWL API (Horridge and Bechhofer 2011).

[Figure 11 near here]

Figure 12a shows the results of amalgamating the accident clusters into polygons and collapsing and then pruning the road network. The road pruning was performed by respecting the spatial relationships between roads and accidents described earlier. This led to the retention of some minor roads that might normally have been removed. In Figure 12b the road network was generalized in the same manner but the accidents have been aggregated i.e. clusters of accidents have been replaced by points that vary in radius with the size of the clusters. In instances where OWL queries did not result in definitive answers (because the ontology was not sufficiently complete), the user was required to make a choice between candidate operators.

[Figure 12 near here]

Extending the Ontology

The representation of declarative knowledge simplifies the task of expanding the ontology to include other use cases. For example, much of the knowledge required to map crimes and accidents, both punctual events, is common to both. This was tested by creating a new *PunctualEventFeatureType* and adding to it the existing *AccidentFeatureCollection* class and a new *CrimeFeatureType* class. Common properties, such as geometry, are now defined in the *PunctualEventFeatureType* class. The mapping of the road accidents is unchanged and when the crimes are mapped, the crimes are generalised in a similar manner to the accidents. However, no road network is mapped since no semantic relation between crimes and roads has been declared. A new context, such as buildings, will be required. In summary, the requirement to map a new type of feature does not necessarily mean the addition of a complete description of that new feature to the ontology if some of its characteristics can be abstracted to a super-class.

Conclusions and further work

Encapsulating cartographic knowledge explicitly and formally in an ontology allows it be shared, expanded and utilized in on-demand mapping systems. This approach enables us to reason about the cartographic design process in a way not previously possible. By encapsulating cartographic knowledge as declarative knowledge we can extend beyond that which is explicitly represented using inference (Genesereth and Nilsson 1998). The ambition of this ontology has been to utilize statements about geographic features in order to support on-demand mapping.

One can argue about the quality of the solutions. Indeed as it stands the definition of the individual operator concepts requires greater detailing with better incorporation of the semantics of design. However, this has been the first attempt at formalizing concepts such as the generalization operator thus enabling us to move away from natural language descriptions. In the next phase, the ontology will be expanded in order to manage further use cases. The knowledge required to model a single use case, road accidents, is extensive and it may seem that the task of developing a comprehensive ontology is overwhelming. However, the representation of declarative knowledge should ease that task. The advantage of ontologies is that they can create a shared understanding of the map generalisation process and can be used to approach this process from a semantic or geographic perspective, rather than just a geometric one. Ontologies can also make explicit the concepts and relationships in a form that supports machine learning, which is particularly important for on-demand mapping. Ontologies alone will not solve the problem of on-demand mapping but instead are a good candidate for representing the knowledge required for on-demand mapping.

The definition of an ontology includes the principle of a *shared* conceptualization and now that the techniques for building a generalization ontology have been refined and experience gained it will be necessary to collaborate with domain experts to further develop the ontology. Collaborative tools such as webProtégé (Tudorache et al. 2013) are considered to be the way forward in this regard. The task may also be made easier by incorporating existing work such as ontology design patterns for topographic feature types (Varanka 2011) and spatial relations (Touya et al. 2014). It has been recognized that for successful multi-scale mapping it is necessary to capture the semantics of geographic features and the relationships between them (Stoter et al. 2010; Wolf 2009; Dutton and Edwardes 2006) but it is also necessary to capture the characteristics of the processes of generalization. Over twenty years ago, Nyerges (1991) recognized that cartographers lacked the means to systematically document the knowledge required for generalization; ontologies now provide such a tool.

Acknowledgements

We are most grateful for the part funding of this research by the Ordnance Survey of Great Britain.

References

- Balley, S. and N. Regnauld. 2011, "Models and standards for on-demand mapping." In *25th International Cartographic Conference*. Paris, France.
- Beard, M.K. 1991. "Constraints on rule formation." In *Map Generalization: Making rules for knowledge representation*, edited by B. Battenfield and R. B. McMaster. New York: Longman Scientific & Technical New York.
- Burger, T. and Simperl, E. 2008. Measuring the Benefits of Ontologies in *On the Move to Meaningful Internet Systems: OTM 2008 Workshops Lecture Notes in Computer Science* Volume 5333, 2008, pp 584-594

- Burghardt, D., S. Schmid, and J. Stoter. 2007. "Investigations on Cartographic Constraint Formalisation." In *10th Workshop of the ICA commission on Generalization and Multiple Representation*. Moscow, Russia.
- Cecconi, A. 2003. "Integration of cartographic generalization and multi-scale databases for enhanced web mapping." University of Zurich.
- Davis, R., H. Shrobe, and P. Szolovits. 1993. "What is a knowledge representation?" *AI Magazine* 14 (1).
- De Nicola, A., M. Missikoff, and R. Navigli. 2009. "A software engineering approach to ontology building." *Information Systems* 34 (2):258-75.
- Dutton, G., and A. Edwardes. 2006. "Ontological Modeling of Geographical Relationships for Map Generalization." In *Workshop of the ICA Commission on Map Generalization and Multiple Representation*. Portland, USA.
- ESRI. "GIS Dictionary." Accessed 2nd January 2015.
<http://support.esri.com/en/knowledgebase/Gisdictionary>
- Fisher, P.F., and W. A. Mackaness. 1987. "Are cartographic expert systems possible?" In *Proceedings AutoCarto 8*, 530-4.
- Foerster, T., J. Stoter, and B. Köbben. 2007. "Towards a formal classification of generalization operators." In *International Cartographic Conference 2007*. Moscow.
- Genesereth, M.R., and N.J. Nilsson. 1998. *Logical foundations of artificial intelligence*. Palo Alto, California: Morgan Kaufmann Publishers.
- Grüninger, M., and M. Fox. 1995. "Methodology for the Design and Evaluation of Ontologies." In *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal.
- Harrie, L., and R. Weibel. 2007. "Modelling the Overall Process of Generalization." In *Generalization of Geographic Information*, edited by W. A. Mackaness, A. Ruas and L. Sarjakoski, 67-87. Amsterdam: Elsevier Science B.V.
- Horridge, M., and S. Bechhofer. 2011. "The OWL API: A Java API for OWL ontologies." *Semantic Web* 2 (1):11-21.
- International Cartographic Association. 1973. *Multilingual Dictionary of Technical Terms in Cartography*. Franz Steiner Verlag.
- Iqbal, R., M. A. A. Murad, A. Mustapha, and N. M. Sharef. 2013. "An Analysis of Ontology Engineering Methodologies: A Literature Review." *Research Journal of Applied Sciences, Engineering and Technology* 16 (6).
- Kavouras, M., and M. Kokla. 2008. *Theories of geographic concepts : ontological approaches to semantic integration*. Boca Raton: CRC Press.

- Kilpeläinen, T. 2000. "Knowledge Acquisition for Generalization Rules." *Cartography and Geographic Information Science* 27 (1).
- Kulik, L., M. Duckham, and M. Egenhofer. 2005. "Ontology-driven map generalization." *Journal of Visual Languages & Computing* 16 (3):245-67.
- Lamy, S., A. Ruas, Y. Demazeau, M. Jackson, W. A. Mackaness, and R. Weibel. 1999. "The application of agents in automated map generalization." In *19th International Cartographic Conference*. Ottawa.
- Lecordix, F., and C. Lemarie. 2007. "Managing Generalization Updates in IGN Map Production." In *Generalization of Geographic Information*, edited by W.A. Mackaness, A. Ruas and L. T. Sarjakoski, 285-300. Amsterdam: Elsevier Science B.V.
- Lüscher, P., R. Weibel, and W. A. Mackaness. 2008. "Where is the Terraced House? On the Use of Ontologies for Recognition of Urban Concepts in Cartographic Databases." In *Headway in Spatial Data Handling*, edited by A. Ruas and C. Gold, 449-66. Berlin/Heidelberg: Springer.
- McMaster, R. 1991. "Conceptual frameworks for geographical knowledge." In *Map Generalization: Making rules for knowledge representation*, edited by B. Buttenfield and R. B. McMaster. New York: Longman Scientific & Technical New York.
- McMaster, R. B., and K. S. Shea. 1992. *Generalization in digital cartography*. Washington D.C.: Association of American Geographers.
- Monmonier, M. S. 1984. "Geographic Information and Cartography." *Progress in Human Geography* 8 (3):381-91
- Morrison, J. L. 1997. "Topographic mapping in the twenty-first century." In *Framework for the World*, edited by D. Rhind, 14-27. Cambridge, England: Geoinformation International.
- Nickerson, B.G. 1991. "Knowledge engineering for generalization." In *Map Generalization: Making rules for knowledge representation*, edited by B. Buttenfield and R. B. McMaster. New York: Longman Scientific & Technical New York.
- Noy, N.F. , and D. McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." In.: Technical Report SMI-2001-0880, Stanford University.
- Nyerges, T. L. 1991. "Representing geographical meaning." In *Map Generalization: Making rules for knowledge representation*, edited by B. Buttenfield and R. B. McMaster. New York: Longman Scientific & Technical.
- Oxford English Dictionary. 2014a. "event, n.": Oxford University Press.
- Oxford English Dictionary. 2014b. "symptom, n.": Oxford University Press.

- Revell, P., N. Regnauld, and B. Bulbrooke. 2011. "OS VectorMap District: automated generalization, text placement and conflation in support of making public data public." In *25th International Cartographic Conference*. Paris.
- Rich, E., and K. Knight. 1991. *Artificial intelligence*. 2nd ed. New York: McGraw-Hill.
- Roth, R., C. Brewer, and M. Stryker. 2011. "A typology of operators for maintaining legible map designs at multiple scales." *Cartographic Perspectives* 68.
- Sarjakoski, L. Tiina. 2007. "Conceptual Models of Generalization and Multiple Representation." In *Generalization of Geographic Information*, edited by W. A. Mackaness, Anne Ruas and L. Tiina Sarjakoski, 11-35. Amsterdam: Elsevier Science B.V.
- Sommer, S., and T. Wade. 2006. *A to Z GIS: An Illustrated Dictionary of Geographic Information Systems*: Esri Press.
- Stanford Center for Biomedical Research. "Protégé." Accessed 2nd January 2015. <http://protege.stanford.edu/>.
- Steiniger, S., and R. Weibel. 2007. "Relations among Map Objects in Cartographic Generalization." *Cartography and Geographic Information Science* 34 (3):175-97.
- Stevens, R., and P. Lord. "Managing synonymy in OWL." Accessed 2nd January 2015. <http://ontogenesis.knowledgeblog.org/1236>.
- Stoter, J., M. Post, V. van Altena, R. Nijhuis, and B. Bruns. 2014. "Fully automated generalization of a 1:50k map from 1:10k data." *Cartography and Geographic Information Science* 41 (1):1-13.
- Stoter, J., T. Visser, P. van Oosterom, W. Quak, and N. Bakker. 2010. "A semantic-rich multi-scale information model for topography." *International Journal of Geographical Information Science* 25 (5):739-63. Studer, R., R. Benjamins, and D. Fensel. 1998. "Knowledge engineering: Principles and methods." *Data and Knowledge Engineering* 25 (1-2):168-98.
- Sure, Y., S. Staab, and R. Studer. 2009. "Ontology Engineering Methodology." In *Handbook on Ontologies*, edited by S. Staab and R. Studer. Berlin/Heidelberg: Springer-Verlag.
- Taillandier, P., and F. Taillandier. 2012. "Multi-criteria diagnosis of control knowledge for cartographic generalization." *European Journal of Operational Research* 217 (3):633-42.
- Touya, G., B. Bucher, G. Falquet, K. Jaara, and S. Steiniger. 2014. "Modelling Geographic Relationships in Automated Environments." In *Abstracting Geographic Information in a Data Rich World*, edited by D. Burghardt, C. Duchene and W. A. Mackaness, 53-82. Heidelberg: Springer.
- Tudorache, T., C. Nyulas, N. F. Noy, and M. A. Musen. 2013. "WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the Web." *Semantic Web* 4 (1):89-99.

Uschold, M., M. King, S. Moralee, and Y. Zorgios. 1998. "The Enterprise Ontology." *The Knowledge Engineering Review* 13 (1):31-89.

Varanka, D. 2011. "Ontology Patterns for Complex Topographic Feature Types." *Cartography and Geographic Information Science* 38 (2):126-36.

Varanka, D., and H. Caro. 2013. "Spatial Relation Predicates in Topographic Feature Semantics." In *Cognitive and Linguistic Aspects of Geographic Space*, edited by M. Raubal, D. M. Mark and A. U. Frank, 175-93. Springer Berlin/Heidelberg.

Wilson, D., M. Bertolotto, and J. Weakliam. 2010. "Personalizing map content to improve task completion efficiency." *International Journal of Geographical Information Science* 24 (5):741-60.

Wolf, E. B. 2009. "Ontology-Driven Generalization of Cartographic Representations by Aggregation and Dimensional Collapse." In *The Semantic Web - ISWC 2009*, edited by A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta and K. Thirunarayan, 990-7. Springer Berlin/Heidelberg.

Yan, J., E. Guilbert, and E. Saux. 2015. "An Ontology of the Submarine Relief for Analysis and Representation on Nautical Charts." *The Cartographic Journal* 52 (1):58-66.


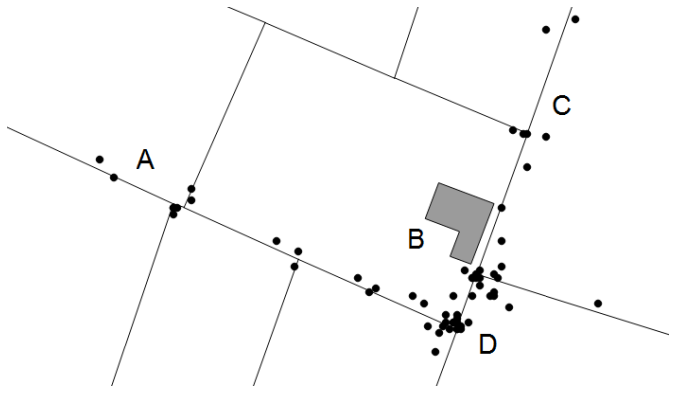
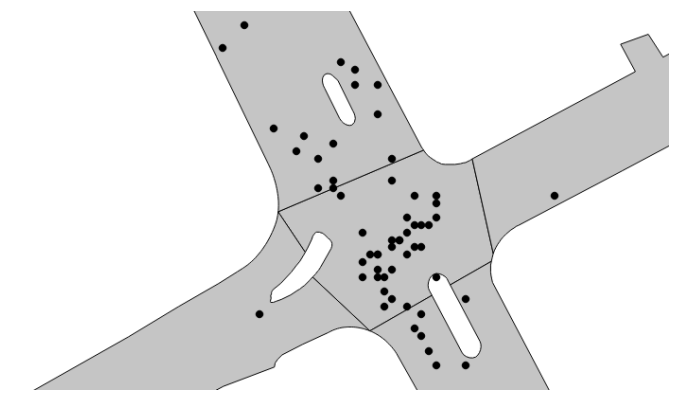
User type	Aim	Relative scale	Example map
(a) Road safety expert	Identify accident ‘hot-spots’ in the city center	Small	
(b) Parent	Identify a safe walking route to school. The school building adds context.	Medium	
(c) Road engineer	Identify problem arm at a junction	Large	

Table 1 Potential users of an on-demand mapping system

Concept	Description
Cartographic generalisation	'The abstraction, reduction, and simplification of <u>features</u> so that a map is clear and uncluttered at a given scale'(Sommer and Wade, 2006). This definition is used in preference to the frequently cited International Cartographic Association (1973) definition: 'the selection and simplified representation of detail appropriate to scale and/or purpose of a map' since it better represents the aims of the use case.
Congestion	The <u>geometric condition</u> where 'too many geographic <u>features</u> need to be represented in a limited physical space on the map' (McMaster and Shea, 1992).
Geometric Condition	Conditions in the mapped <u>features</u> that are caused by a reduction in scale and used to determine the need for <u>generalisation</u> (McMaster and Shea, 1992). For example, <u>congestion</u> .
Event	An incidence or occurrence (Oxford English Dictionary, 2014a); an immaterial object that can share the same space as a <u>topographic feature</u> or another event.
Feature	A mapped object. Can be <i>material</i> (topographic) or <i>immaterial</i> (such as an event). Features can be grouped by <u>feature type</u> . E.g. Buildings.
Feature collection	A set of <u>features</u> all of the same <u>feature type</u> .
Feature type	A class of features e.g. buildings, rivers.
Geometry	'The measures and properties of points, lines, and surfaces. In a GIS, geometry is used to represent the spatial component of geographic <u>features</u> ' (ESRI, 2014).
High Feature Density	A <u>symptom</u> of <u>congestion</u> .
Operator	'Abstract or generic description of the type of modification that can be applied when changing scale'(Roth et al., 2011). An abstract function that transforms geographic data. An operator is implemented by one or more <u>algorithms</u> .
Road accident	A punctual event <u>feature type</u> . Takes place on a <u>road segment</u> .
Road segment	Section of road between two junctions (nodes). A topographic <u>feature type</u> . Part of a <u>network</u> .
Symptom	'A phenomenon or circumstance accompanying some <u>condition</u> ... and serving as evidence of it' (Oxford English Dictionary, 2014b)
Measure algorithm	A procedure for measuring a particular <u>symptom</u> .
Transformation algorithm	A procedure for implementing a particular <u>operator</u> . Some transformation algorithms specialise in particular <u>feature types</u> . Normally termed a <i>generalisation</i> algorithm. Implemented in computer code.

Table 2 Example concept descriptions

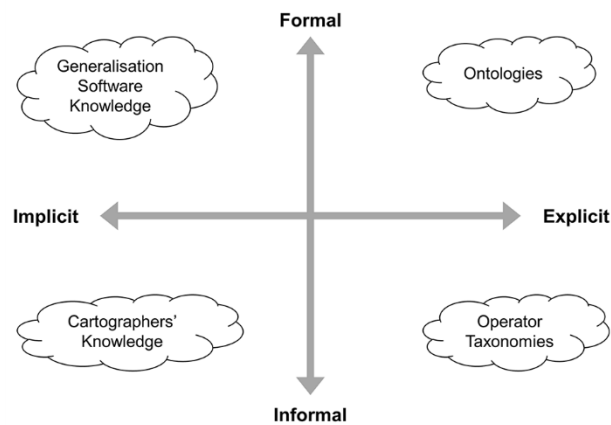


Figure 1 Representing cartographic knowledge

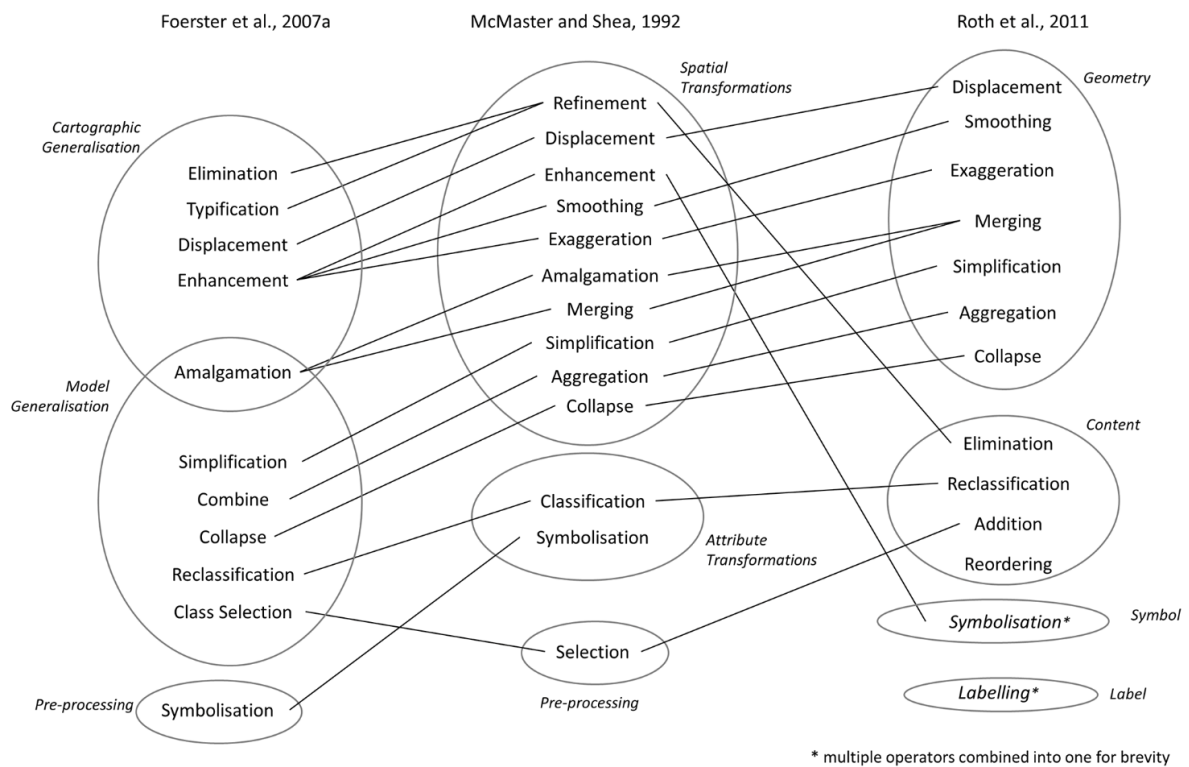


Figure 2 A comparison of three generalization operator taxonomies

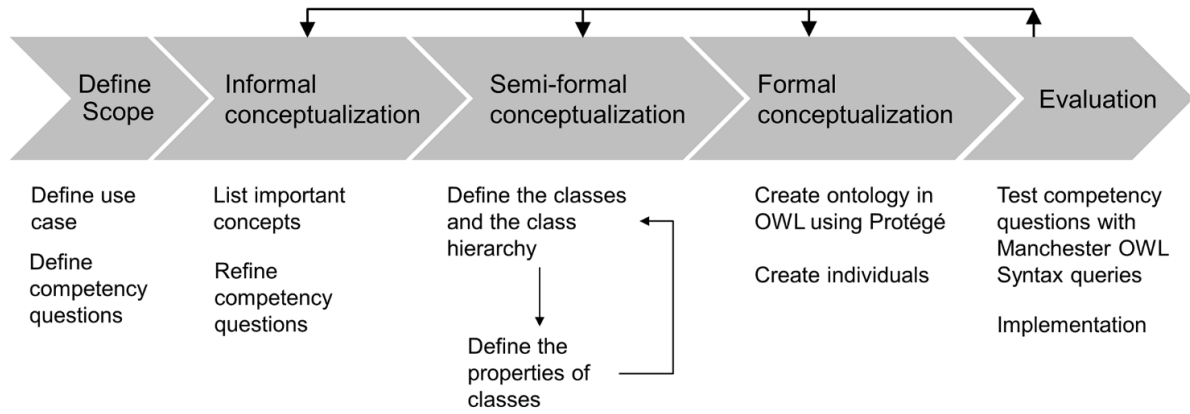


Figure 3 Methodology for building the ontology

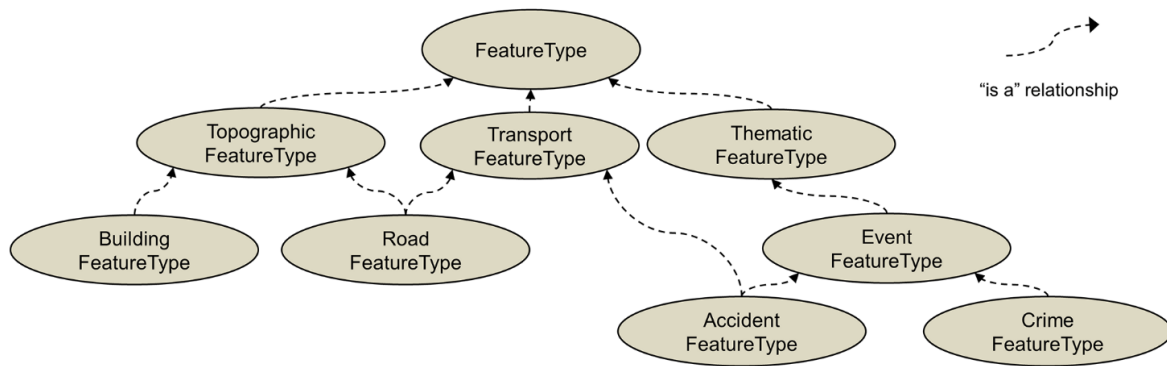


Figure 4 Feature type class hierarchy

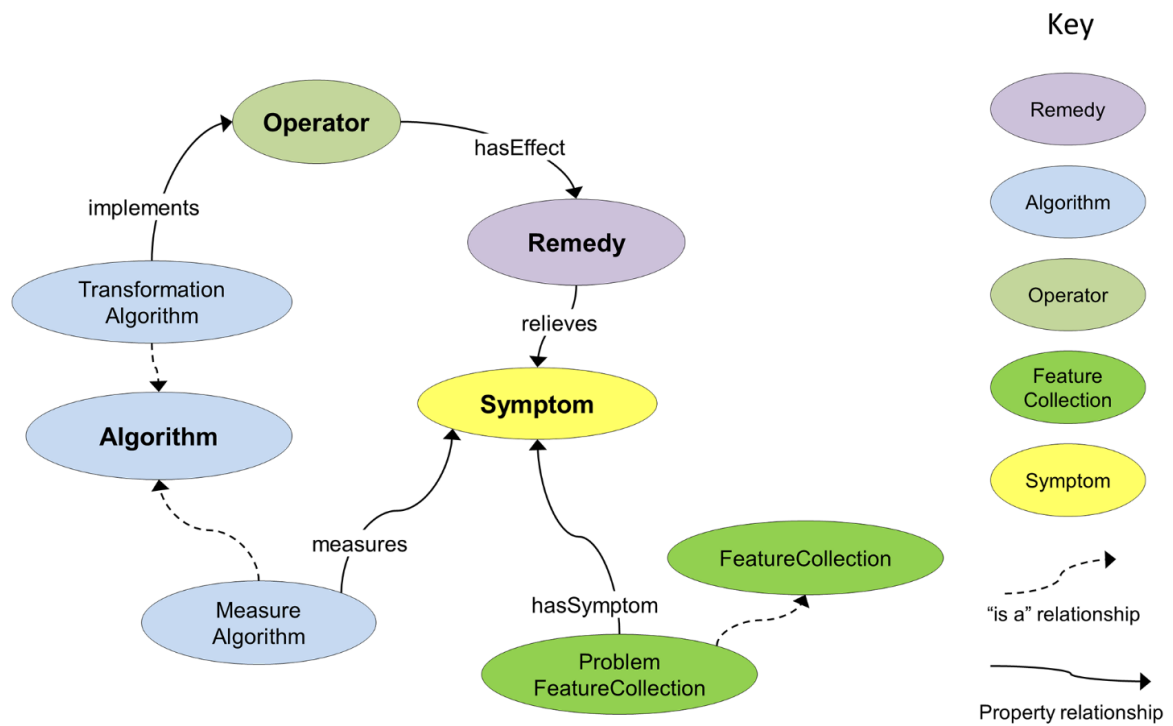


Figure 5 Relationships between the top-level concepts in the ontology

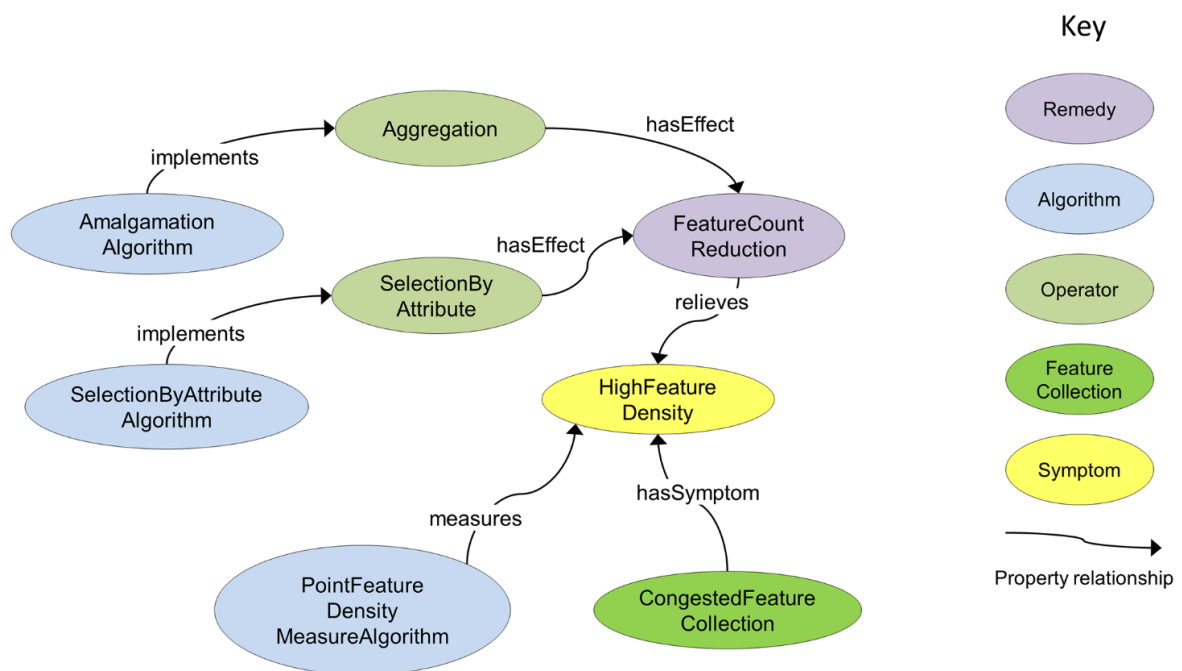


Figure 6 Particular classes for generalizing congested point features

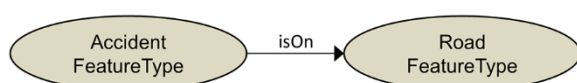


Figure 7 The semantic relation between accidents and roads

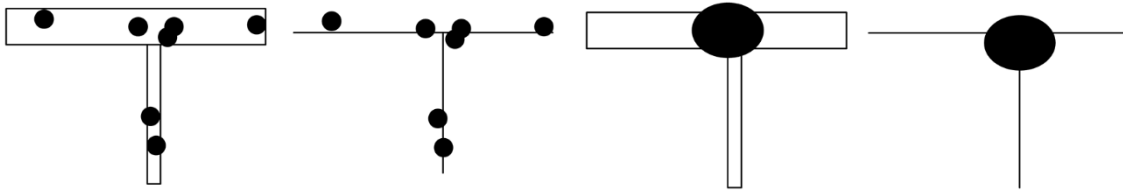


Figure 8 Spatial relation predicates between accidents and roads

- (a) contained by
- (b) adjacent
- (c) intersects
- (d) intersects

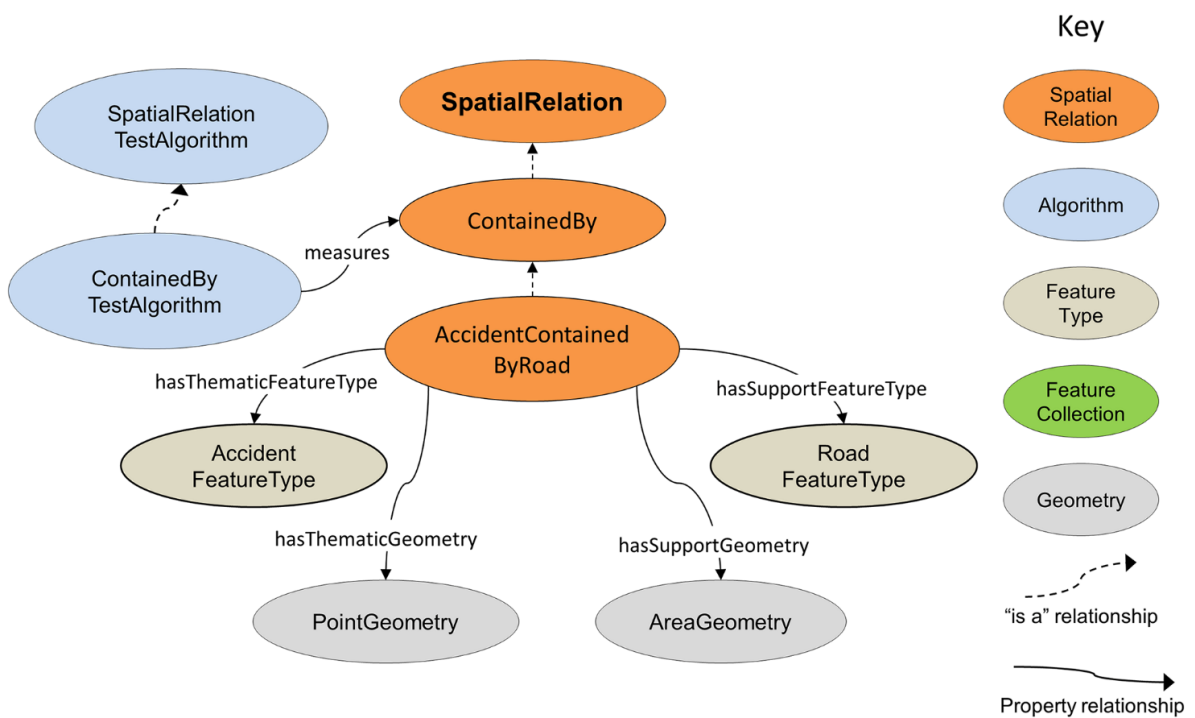


Figure 9 Modelling the 'contained by' spatial relation as it relates to accidents and roads

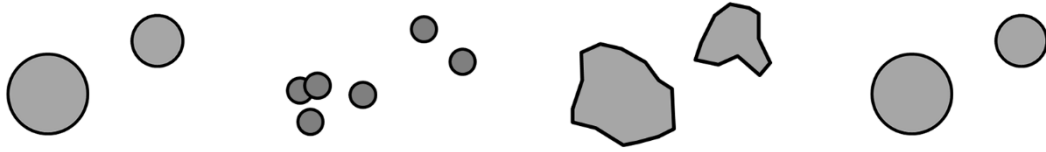


Figure 10 Reducing congestion in point features

- (a) original features
- (b) selection by attribute
- (c) amalgamation
- (d) aggregation

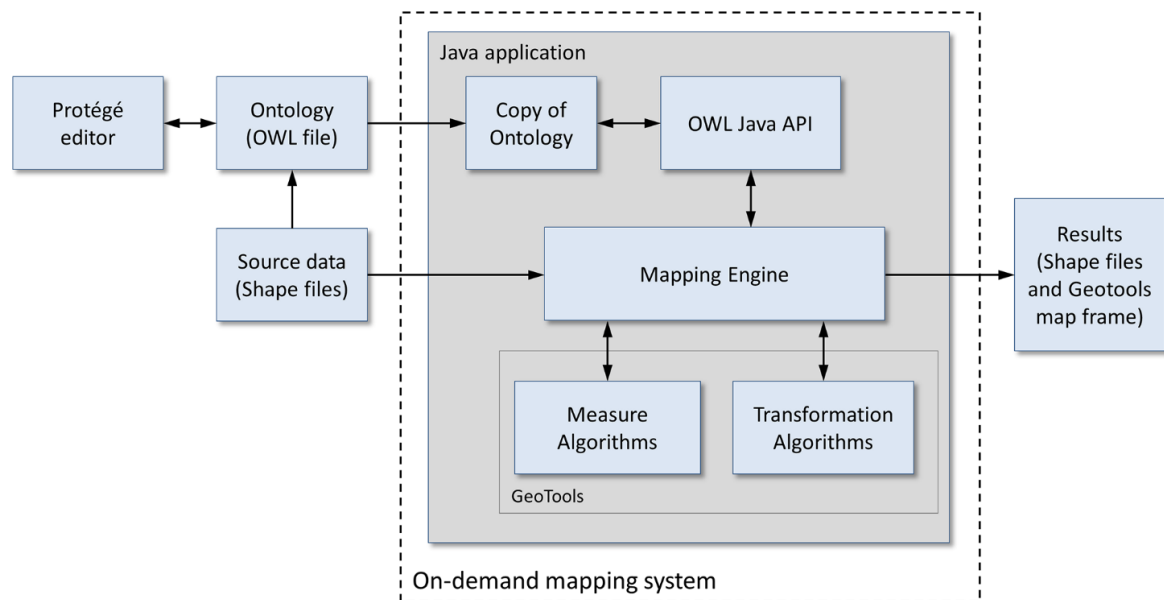


Figure 11 On-demand mapping system design

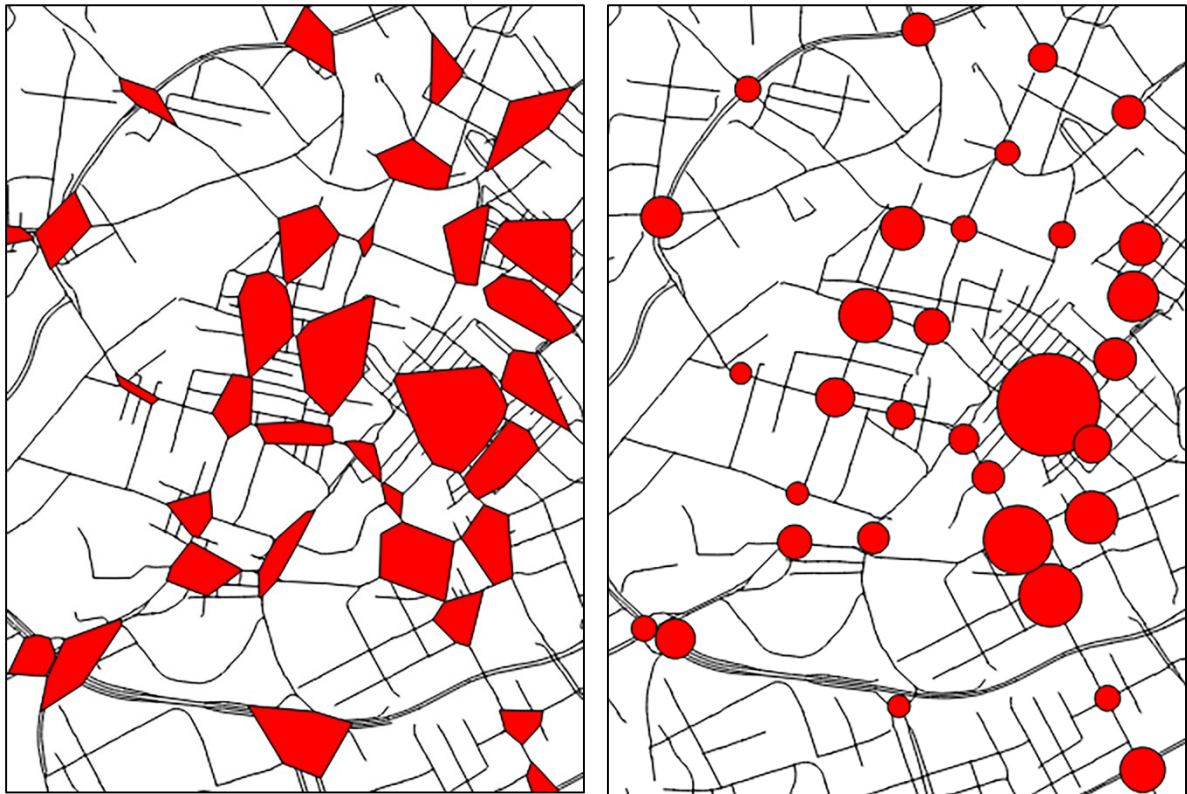


Figure 12 Road accidents in Manchester city center (scale approx. 1:40000)

- (a) Amalgamated accidents
- (b) Aggregated accidents